

BAB III

PERANCANGAN SISTEM

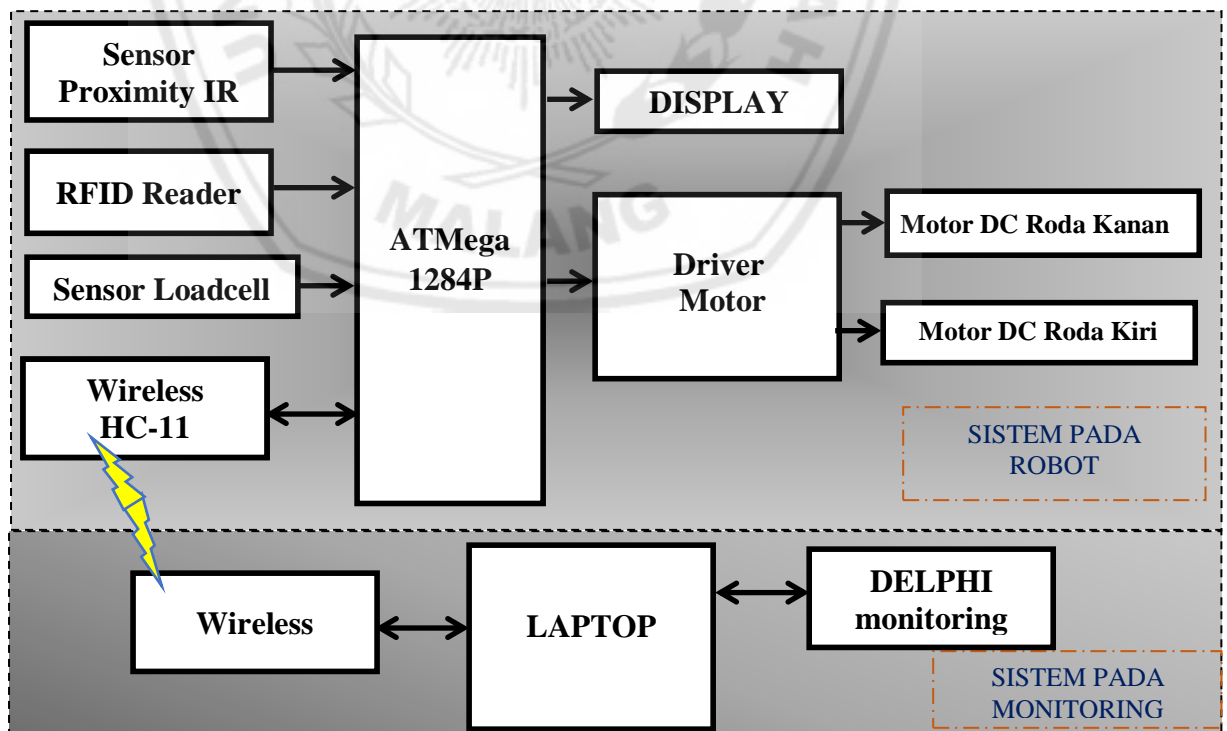
3.1. Tinjauan Umum

Perancangan Sistem Robot *Line Follower* untuk Pembelajaran di Taman Rekreasi Sengkaling dengan Identifikasi RFID disertai *Wireless Monitoring* merupakan media pembelajaran Elektronika Robotika khususnya untuk pemula, hal ini dikarenakan sistem program maupun mekanik yang digunakan cukup sederhana untuk dipahami. Dengan adanya *Wireless Monitoring* robot dapat dipantau dan dikendalikan dengan jarak jauh (radius *wireless*). Agar sistem dapat bekerja dengan hasil yang diharapkan, maka diperlukan perancangan komponen yang dipakai agar kinerja sistem dapat beroperasi optimal.

3.2. Perencanaan Software

Perancangan *software* dari sistem yang akan dirancang mengacu pada Blok Diagram gambar 3.1.

3.2.1. Block Diagram



Gambar 3.1. Blok Diagram Sistem

3.2.2. Prinsip Kerja

Robot pembelajaran dengan identifikasi RFID sebagaimana perancangan menggunakan mekanik dimana mekanik berbentuk miniatur mobil pick-up yang terdapat dua roda aktif yang digerakkan oleh motor dan dua sisi lainnya menggunakan roda bebas. Untuk bagian bawah dijalan landasan terdapat Sensor *Proximity IR* yang digunakan untuk mendeteksi jalur lintasan (*track*) dan RFID *reader* yang bertugas membaca ID di setiap titik kota. Sensor *Loadcell* untuk mendeteksi ada tidaknya benda atau paket yang diangkut.

Pada kondisi saat Robot diaktifkan maka dilakukan inisialisasi untuk melakukan *setting* awal pada mikrokontroller. Inisialisasi ini sendiri berperan untuk memanggil sistem yang akan dipakai. Untuk *input* data lokasi awal, lokasi tujuan, dan ataupun pengangkutan paket terdapat pada aplikasi.

Seanjutnya sistem pada robot menunggu perintah yang akan di *input*, apabila robot belum menerima data dari aplikasi maka robot akan berjalan sebagaimana layaknya Robot *Line Follower* biasa yang hanya mengikuti garis track. Setelah perintah diberikan maka data yang dimasukkan pada aplikasi akan dikirimkan pada robot melalui *wireless* HC-12. Saat perintah diterima pada robot, maka robot akan menampilkan kembali data yang diterima tersebut pada LCD. Setelah data diterima dan ditampilkan kembali pada robot, maka robot akan mulai bergerak dengan dipandu oleh sensor *proximity ir* dan sensor RFID. Robot akan terus bergerak mengikuti jalur hingga menemui titik kota, apabila robot menemukan titik kota maka robot tersebut akan berhenti sejenak untuk menentukan apakah kota tersebut merupakan tujuannya apa bukan. Apabila kota tersebut bukan tujuannya maka robot akan kembali bergerak, namun apabila kota tersebut tujuannya maka robot akan menampilkan apa yang harus dilakukan.

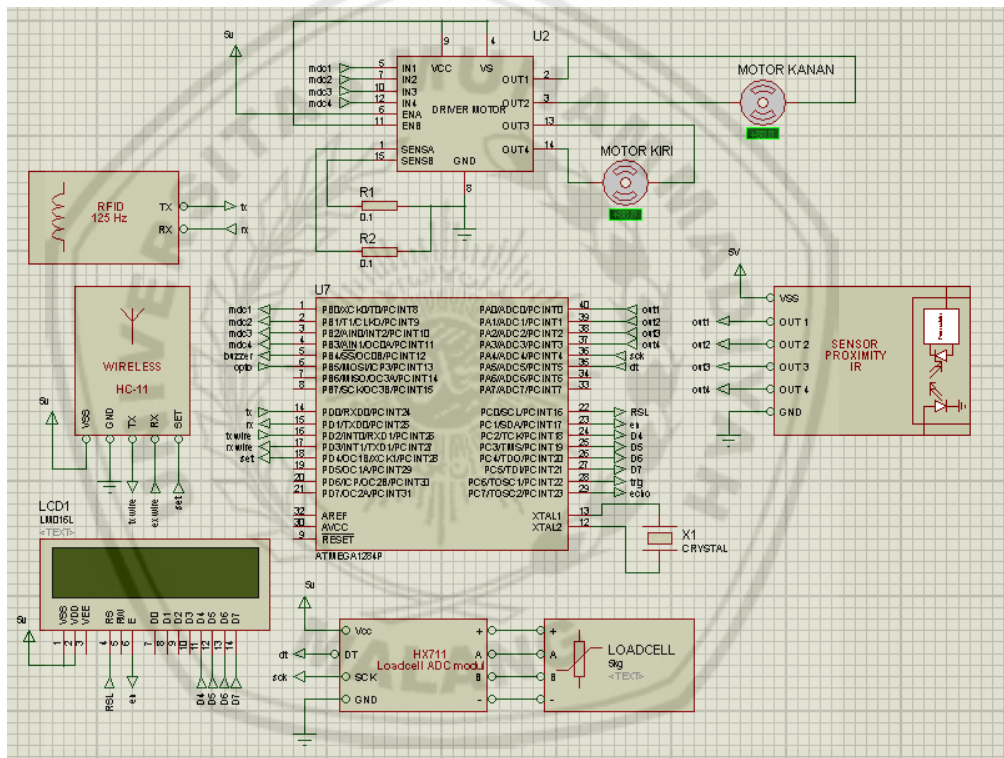
Disaat robot menemukan jalur percabangan maka robot akan berhenti sejenak untuk menentukan kearah mana robot tersebut berbelok. Robot akan berhenti disaat telah mencapai tujuan akhir dari data yang telah diberikan di awal (pada aplikasi) yang telah ditentukan serta kondisinya telah terselesaikan dan robot akan menampilkan “turunkan paket” dan robot kembali *stanby* menunggu perintah selanjutnya.

3.3. Perancangan Software

3.3.1. Pengaturan Mikrokontroler ATmega 1284P

Mikrokontroler merupakan salah satu komponen utama dalam perancangan yang berperan sebagai kontrol pada Robot, oleh karena itu pengaturan serta pemetaan pin yang ada pada mikrokontroler terhadap pin dari komponen *hardware* yang berintegrasi haruslah diperhatikan.

Fungsi dari setiap pin maupun port pada mikrokontroler berbeda, sesuai dengan kebutuhan sistem kerja pada alat. Pada gambar 3.2. berikut merupakan fungsi dari pin ataupun port pada mikrokontroler serta komponen *hardware* lainnya.



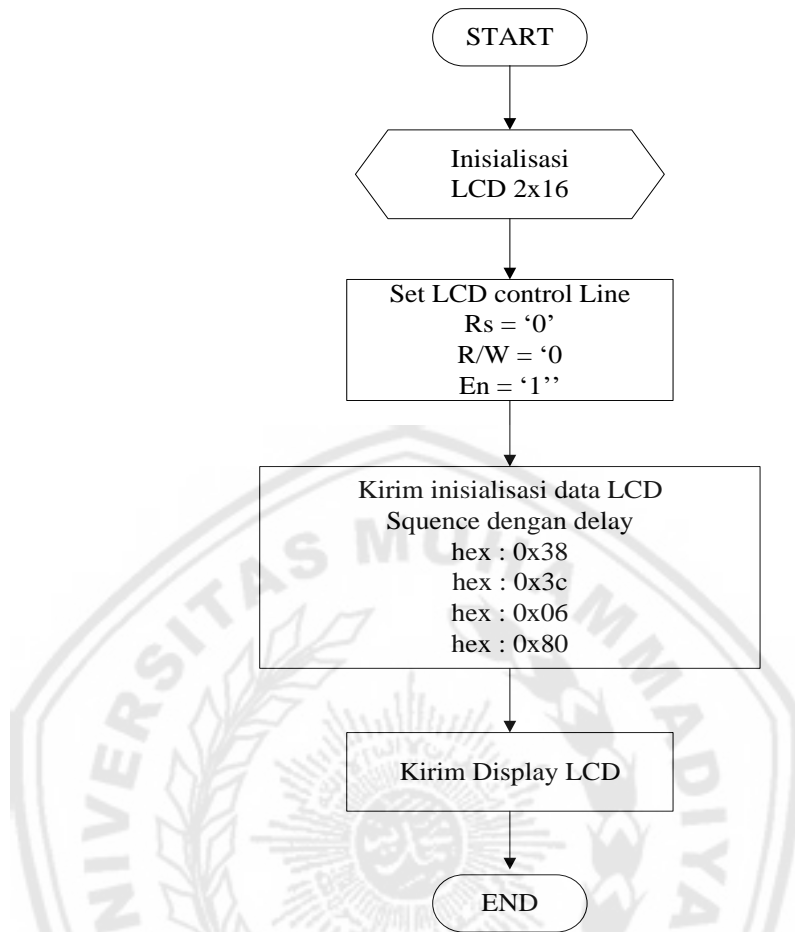
Gambar 3.2. Susunan pin Mikrokontroler dan *hardware* lainnya

Dilihat dari gambar 3.2. bahwa tidak semua pin pada mikrokontroler digunakan, untuk lebih jelasnya mengenai pin pada mikrokontroler yang digunakan dapat dilihat pada tabel 3.1.

Tabel 3.1. Konfigurasi pin-pin mikrokontroler Atmega1284P

No. Pin Mikrokontroler	Port	No. pin komponen	Nama komponen
1	PB0	IN1	Motor Servo
2	PB1	IN2	Motor Servo
3	PB2	IN3	Motor Servo
4	PB3	IN4	Motor Servo
6	PB5	Mosi	Downloader
7	PB6	Miso	Downloader
8	PB7	SCK	Downloader
9	Reset	reset	Downloader
10	VCC	Saklar	Saklar
11	Ground	Gnd Batt	Battrey
12	x-tal1	-	Cyrstal
13	x-tal2	-	Cyrstal
14	PD0	Tx	RFID
15	PD1	Rx	RFID
16	PD2	tx wire	Wireless HC-11
17	PD3	rx wire	Wireless HC-11
18	PD4	Set	Wireless HC-11
20	PD6	-	-
21	PD7	-	-
22	PC0	OUT0	Proximity IR
23	PC1	OUT1	Proximity IR
24	PC2	Db7	LCD
25	PC3	-	-
26	PC4	Db5	LCD
27	PC5	Db4	LCD
28	PC6	E	LCD
29	PC7	Rs	LCD
30	VCC	Batt	Power Supply
31	GND	ground	Ground
32	AREF	-	-
33	PA7	-	-
34	PA6	-	-
35	PA5	SCK	Loadcell
36	PA4	DT	Loadcell
37	PA3	OUT0	Proximity IR
38	PA2	OUT3	Proximity IR
39	PA1	OUT4	Proximity IR
40	PA0	OUT5	Proximity IR

3.3.2. LCD 2x16



Gambar 3.3. Diagram Alir *Liquid Crystal Display* (LCD) 2x16

Pada gambar 3.3. dapat terlihat bahwa LCD hanya memerlukan inisial awal saja. Sehingga untuk yang selanjutnya LCD hanya perlu memanggil program dan mengisi apa yang perlu ditampilkan pada LCD tersebut. Untuk *list program* LDC dapat dilihat pada gambar 3.4. berikut,

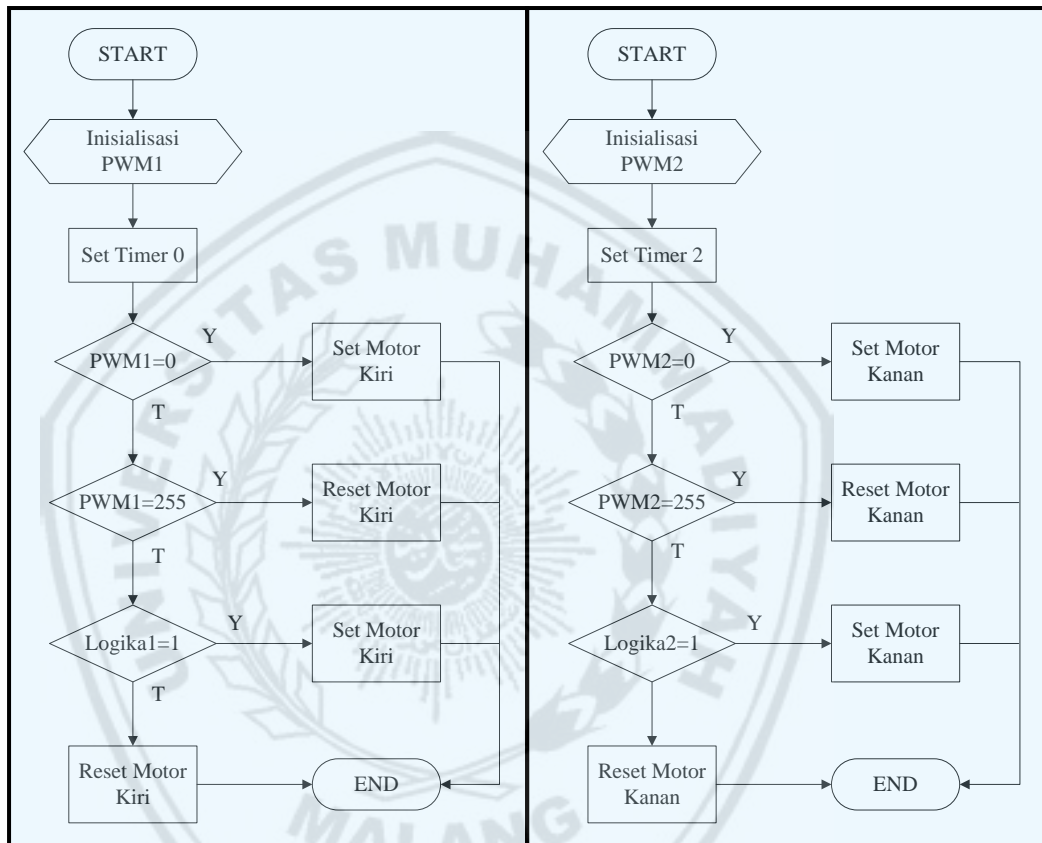
```
$regfile = "m1284pdef.dat"
$crystal = 8000000
$baud = 9600
$baudl = 9600
Config Lcdpin = Pin , Db4 = Portc.5 , Db5 = Portc.4 ,
    Db6 = Portc.3 , Db7 = Portc.2 , E = Portc.6 , Rs
    = Portc.7
Config Lcd = 16 * 2
Lcd "
```

Gambar 3.4. Sript program inisialisasi LCD 2x16

3.3.3. PWM

Tabel 3.2. Perencanaan Kecepatan Motor servo

Kondisi	Kondisi Script	Kecepatan
Normal	0	50 rpm
Low	155	25 rpm
Stop	255	0 rpm



Gambar 3.5. Diagram Alir PWM1 (kiri) dan PWM2 (kanan)

PWM atau yang bisa disebut *Pulse Width Modulation*, merupakan teknik modulasi dengan mengubah lebar pulsa namun masih dengan nilai amplitudo dan frekuensi yang tetap. PWM sendiri disini sebagai pengatur kecepatan pada motor, untuk tiap motor menggunakan PWM sendiri. Pada gambar 3.2. merupakan diagram alir yang digunakan untuk pengaturan kecepatan pada motor kanan dan kiri. Untuk *list program* dapat dilihat pada gambar 3.6. berikut,

```

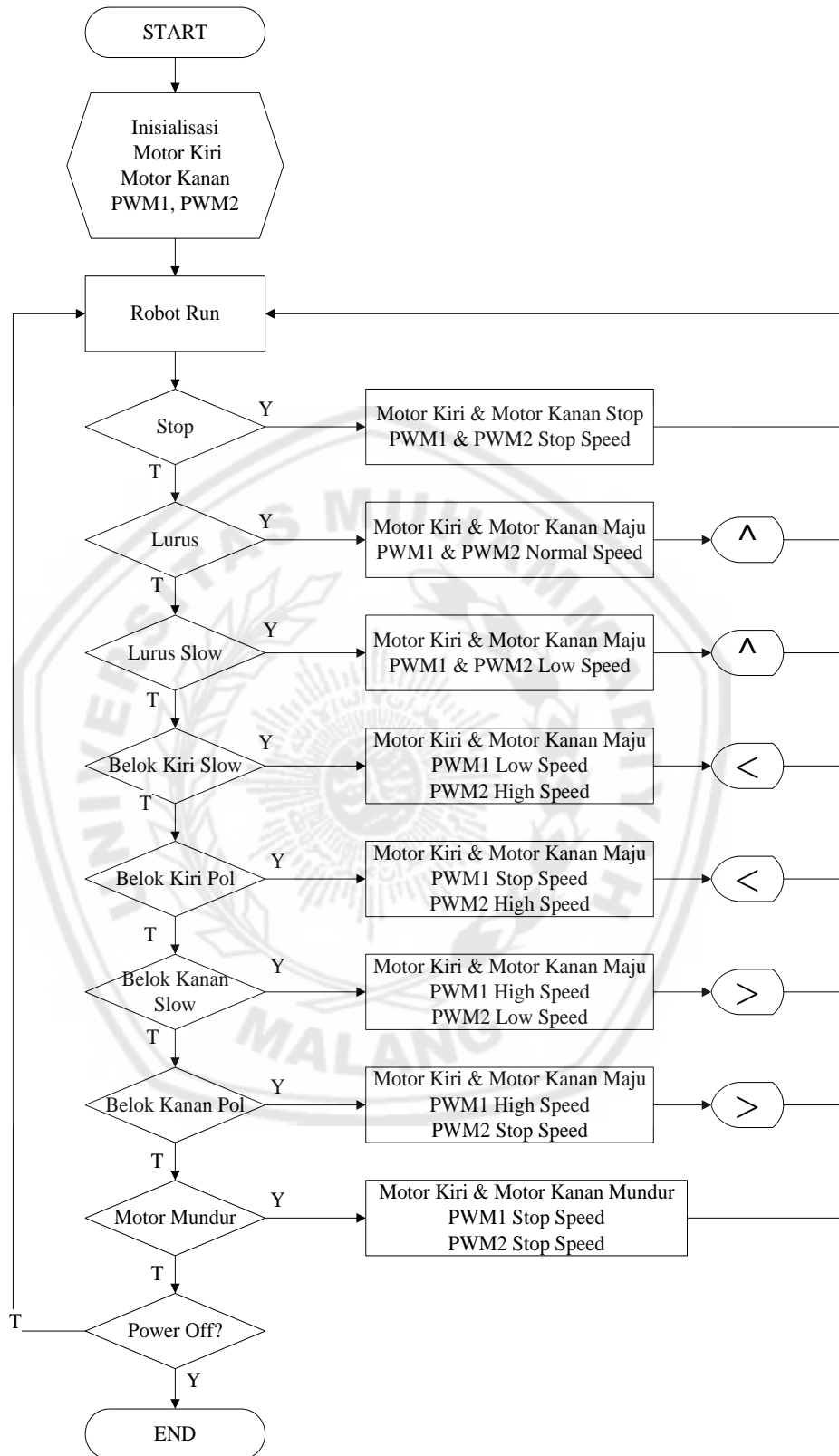
$regfile = "m1284pdef.dat"
$crystal = 8000000
$baud = 9600
$baud1 = 9600
$hwstack = 255      ' default use 32 for the hardware stack
$swstack = 255      ' default use 10 for the SW stack
  Stop_speed Alias   255
  Low_speed Alias    180
  Normal_speed Alias 55
  High_speed Alias   0
  In1 Alias Portb.0
  In2 Alias Portb.1
  In3 Alias Portb.2
  In4 Alias Portb.3
  Motor1 Alias Portb.0
  Motor2 Alias Portb.2
  Config Timer0 = Timer , Prescale = 64
  On 0vf0 Timer_pwm1
  Tcnt0 = 0
  Config Timer2 = Timer , Prescale = 64
  On 0vf2 Timer_pwm2
  Tcnt2 = 0

Timer_pwm1:
  Stop Timer0
  If Status1 = True Then
    If Pwm1 = 0 Then
      Set Motor1
    ElseIf Pwm1 = 255 Then
      Reset Motor1
    ElseIf Logika1 = 1 Then
      Tcnt0 = Pwm1
      Set Motor1
    Else
      Tcnt0 = 255 - Pwm1
      Reset Motor1
    End If
    Toggle Logika1
  Else
    Set Motor1
  End If
  Start Timer0
Return
Timer_pwm2:
  Stop Timer2
  If Status2 = True Then
    If Pwm2 = 0 Then
      Set Motor2
    ElseIf Pwm2 = 255 Then
      Reset Motor2
    ElseIf Logika2 = 1 Then
      Tcnt2 = Pwm2
      Set Motor2
    Else
      Tcnt2 = 255 - Pwm2
      Reset Motor2
    End If
    Toggle Logika2
  Else
    Set Motor2
  End If
  Start Timer2
Return

```

Gambar 3.6. Script program dari PWM

3.3.4. Motor



Gambar 3.7. Diagram Alir Motor

Pada gambar 3.7. menjelaskan bahwa program tersebut dirancang untuk dipanggil pada perintah program yang akan digunakan selanjutnya. Dengan menggunakan PWM, maka motor dapat ditentukan kecepatannya di awal. Sehingga saat akan berbelok halus dan berbelok tajam memiliki kecepatan perputaran yang berbeda. Saat robot diperintahkan untuk berhenti, maka tidak perlu memerintahkan 2 motor satu persatu untuk berhenti. Hanya perlu memanggil perintah program yang telah dibuat maka secara otomatis kedua perintah bisa menjadi satu yaitu Motor Stop, maka kedua roda akan berhenti. Untuk *list program* dari Motor dapat dilihat pada gambar 3.8. sampai gambar 3.10. berikut,

```

$regfile = "m1284pdef.dat"
$crystal = 8000000
$baud = 9600
$baudl = 9600
$hwstack = 255      ' default use 32 for the hardware stack
$swstack = 255      ' default use 10 for the SW stack

Stop_speed Alias    255
Low_speed  Alias    180
Normal_speed Alias  55
High_speed Alias     0

In1 Alias Portb.0
In2 Alias Portb.1
In3 Alias Portb.2
In4 Alias Portb.3

Motor1 Alias Portb.0
Motor2 Alias Portb.2

Motor_kanan_in1 Alias Portb.2
Motor_kanan_in2 Alias Portb.3
Motor_kiri_in1  Alias Portb.0
Motor_kiri_in2  Alias Portb.1

Config Motor_kanan_in1 = Output
Config Motor_kanan_in2 = Output
Config Motor_kiri_in1  = Output
Config Motor_kiri_in2  = Output

Roda_lurus:
    Gosub Motor_kanan_maju
    Gosub Motor_kiri_maju
Return

Motor_kiri_mundur:
    Reset Motor_kiri_in1
    Set Motor_kiri_in2
Return

```

Gambar 3.8. Sript program Motor

```

Motor_kiri_maju:
    Set Motor_kiri_in1
    Reset Motor_kiri_in2
Return

Motor_kiri_stop:
    Set Motor_kiri_in1
    Set Motor_kiri_in2
Return

Motor_kanan_stop:
    Set Motor_kanan_in1
    Set Motor_kanan_in2
Return

Motor_kanan_mundur:
    Reset Motor_kanan_in1
    Set Motor_kanan_in2
Return

Motor_kanan_maju:
    Set Motor_kanan_in1
    Reset Motor_kanan_in2
Return

Belok_kiri:
    Gosub Motor_kanan_maju
    Gosub Motor_kiri_stop
Return

Belok_kanan:
    Gosub Motor_kanan_stop
    Gosub Motor_kiri_maju
Return

Motor_stop:
    Pwm1 = Stop_speed
    Pwm2 = Stop_speed
    Reset Motor_kanan_in2
    Reset Motor_kiri_in2
Return

Lurus:
    Pwm1 = Normal_speed
    Pwm2 = Normal_speed
    Reset Motor_kanan_in2
    Reset Motor_kiri_in2
Return

Lurus_slow:
    Pwm1 = Low_speed
    Pwm2 = Low_speed
    Reset Motor_kanan_in2
    Reset Motor_kiri_in2
Return

Belok_kiri_pol:
    Pwm1 = Stop_speed
    Pwm2 = High_speed
    Reset Motor_kanan_in2
    Reset Motor_kiri_in2
Return

```

Gambar 3.9. Sript program Motor (lanjutan)

```
Belok_kiri_slow:
    Pwm1 = Low_speed
    Pwm2 = High_speed
    Reset Motor_kanan_in2
    Reset Motor_kiri_in2
Return

Belok_kanan_slow:
    Pwm1 = High_speed
    Pwm2 = Low_speed
    Reset Motor_kanan_in2
    Reset Motor_kiri_in2
Return

Belok_kanan_pol:
    Pwm1 = High_speed
    Pwm2 = Stop_speed
    Reset Motor_kanan_in2
    Reset Motor_kiri_in2
Return

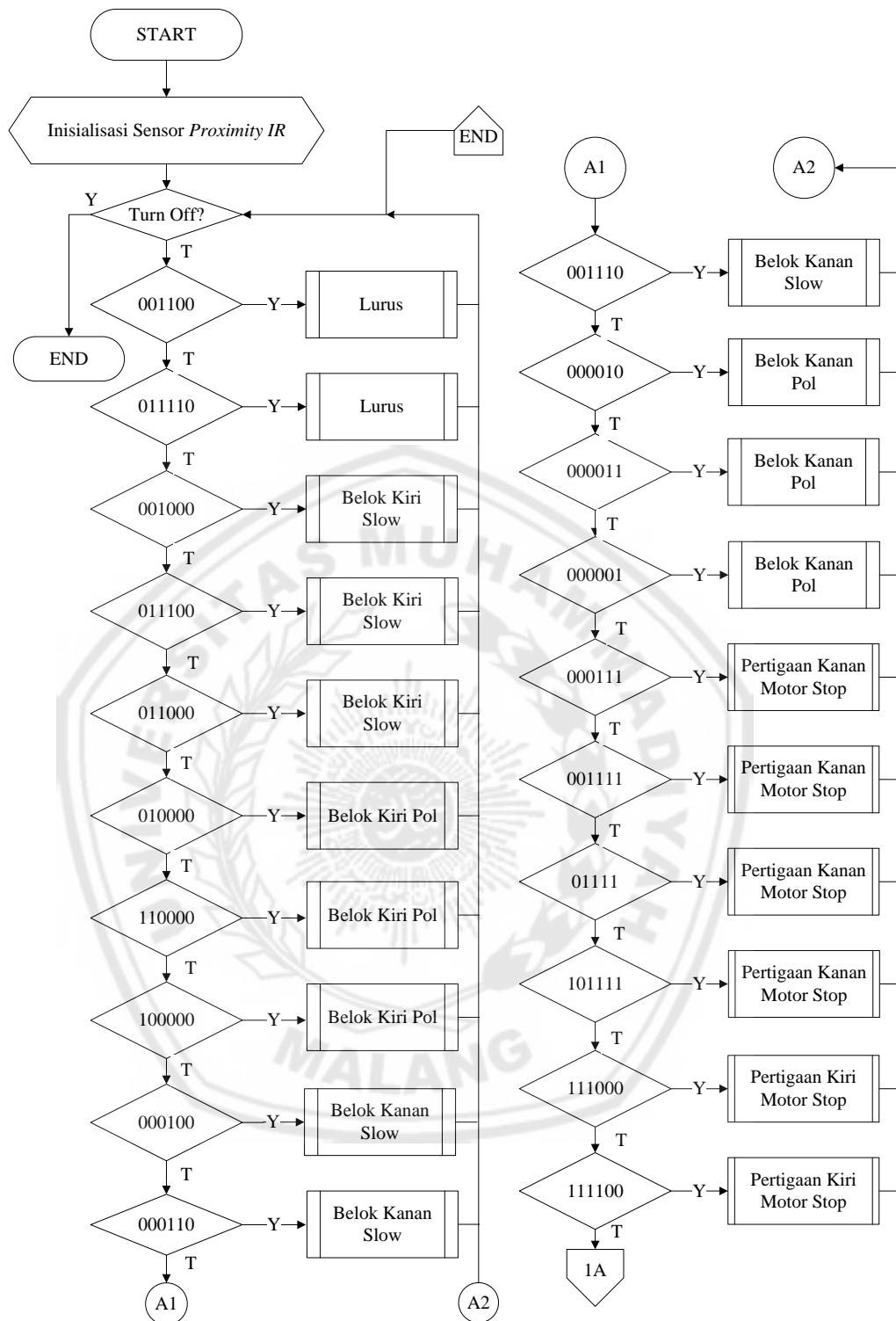
Motor_mundur:
    Pwm1 = Stop_speed
    Pwm2 = Stop_speed
    Set Motor_kanan_in2
    Set Motor_kiri_in2
Return
```

Gambar 3.10. Sript program Motor (lanjutan)

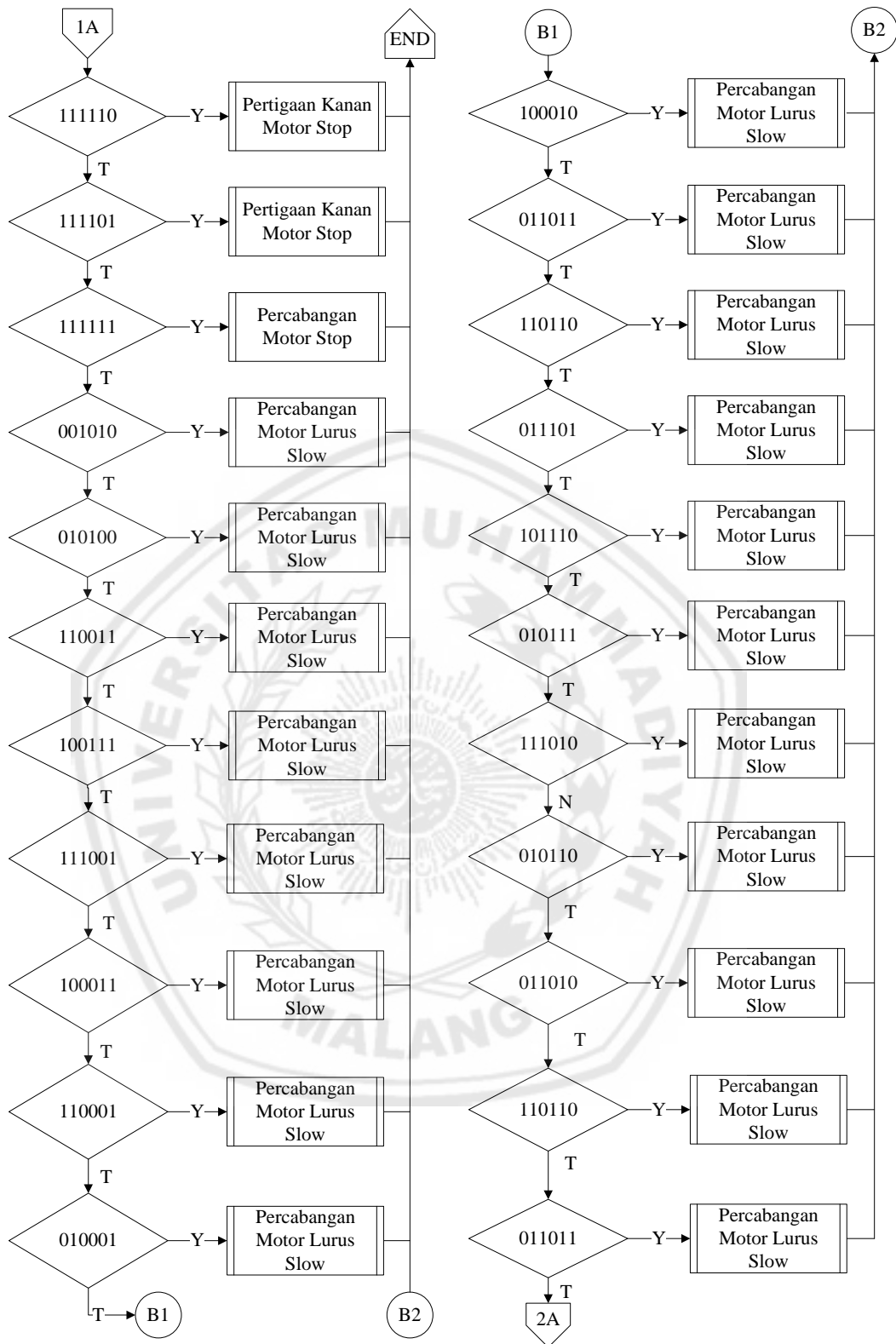
3.3.5. Proximity (IR-transmitter & reading)

Tabel 3.3. K-map dari sensor *Proximity IR*

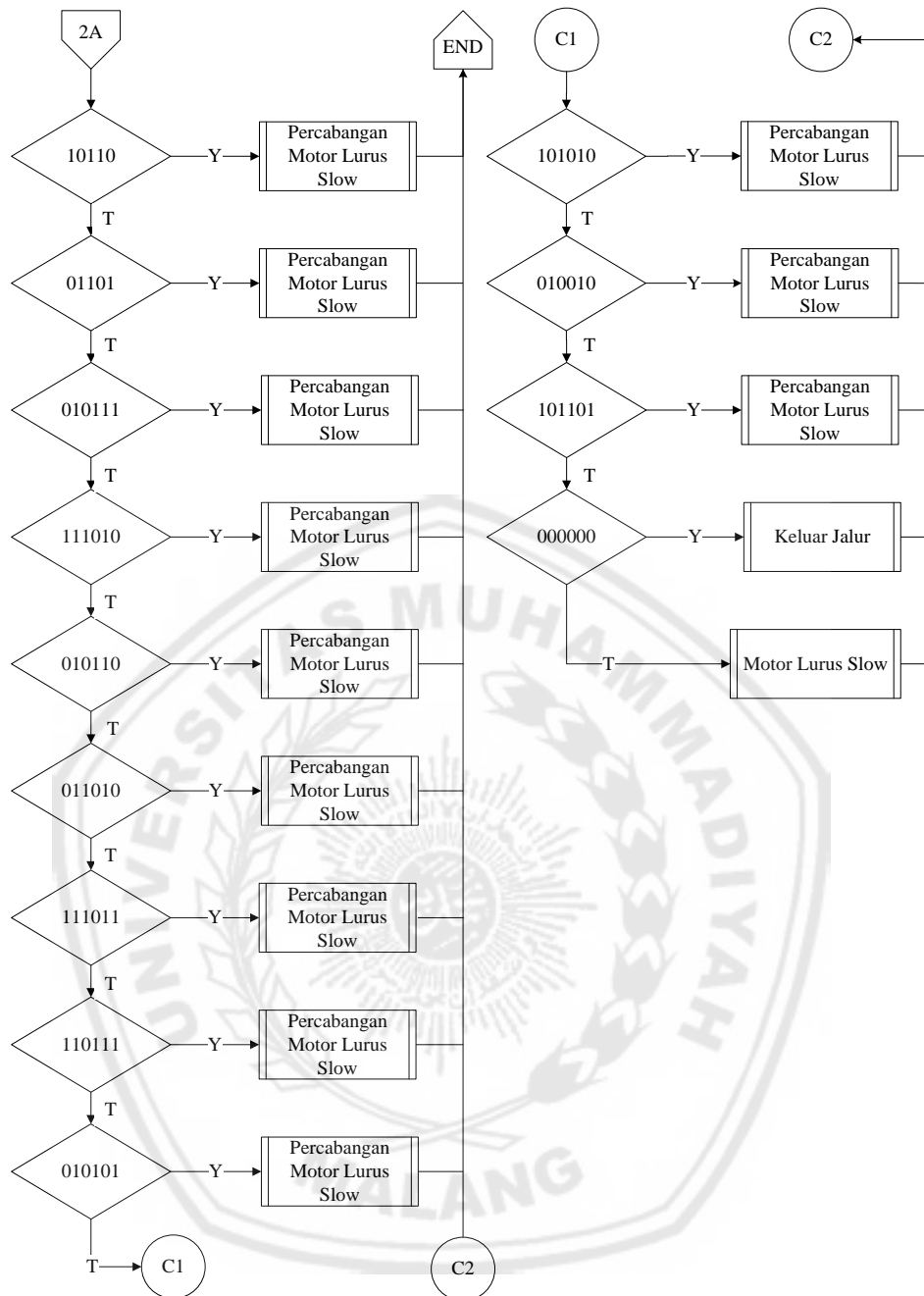
A \ B	000	001	010	011	100	101	111	110
000	Keluar Jalur	Belok Kanan pol	Belok Kanan slow	Belok Kanan pol	Belok Kanan slow		Percabangan Motor Stop	Belok Kanan slow
001	Belok Kiri slow		Percabangan Maju Slow		Jalan Lurus	Percabangan Maju Slow	Percabangan Motor Stop	Belok Kiri slow
010	Belok Kiri slow	Percabangan Maju Slow	Percabangan Maju Slow	Percabangan Maju Slow	Percabangan Maju Slow	Percabangan Maju Slow	Percabangan Maju Slow	Percabangan Maju Slow
011	Belok Kiri slow		Percabangan Maju Slow	Percabangan Maju Slow	Belok Kiri slow	Percabangan Maju Slow	Percabangan Motor Stop	Jalan Lurus
100	Belok Kiri pol		Percabangan Maju Slow	Percabangan Maju Slow			Percabangan Maju Slow	Percabangan Maju Slow
101			Percabangan Maju Slow	Percabangan Maju Slow		Percabangan Maju Slow	Percabangan Maju Slow	Percabangan Maju Slow
111	Percabangan Motor Stop	Percabangan Maju Slow	Percabangan Maju Slow	Percabangan Motor Stop	Percabangan Maju Slow	Percabangan Maju Slow	Percabangan Motor Stop	Percabangan Motor Stop
110	Belok Kiri pol		Percabangan Maju Slow	Percabangan Maju Slow	Percabangan Maju Slow		Percabangan Maju Slow	Percabangan Maju Slow



Gambar 3.11. Diagram Alir reaksi Motor dengan Sensor *Proximity IR*



Gambar 3.12. Diagram Alir reaksi Motor dengan Sensor *Proximity IR*
(lanjutan)



Gambar 3.13. Diagram Alir reaksi Motor dengan Sensor *Proximity IR*
(lanjutan)

Pada gambar 3.11. sampai gambar 3.13. flowchart *Proximity IR* diawali dengan start, selanjutnya inisialisasi sensor yang terhubung pada mikrokontroller. Setelah inisialisasi langkah selanjutnya sensor memulai *scanning* jalur yang akan dilalui. Dan diikuti dengan respon dari pergerakan roda robot, sensor *Proximity IR* akan terus menerus-menerus mengirimkan data masukan ke Mikrokontroler untuk pemantauan keadaan jalur garis yang dilalui. Dengan sensor *Proximity IR* aktif *high*

dan mendeteksi warna gelap, sehingga untuk mendapatkan posisi garis (jalur hitam) ditengah, maka sensor harus menempatkan 2 digit logika 1 (*high*) ditengah. Sehingga apabila adanya tikungan maka sensor akan mendeteksi adanya perubahan dengan menggerakkan pergerakan motor. Untuk *list program* dari *Proximity IR* dapat dilihat pada gambar 3.14. sampai gambar 3.19. berikut.

```

$regfile = "m1284pdef.dat"
$crystal = 8000000
$baud = 9600
$baud1 = 9600

In1 Alias Portb.0
In2 Alias Portb.1
In3 Alias Portb.2
In4 Alias Portb.3
Motor1 Alias Portb.0
Motor2 Alias Portb.2

Motor_kanan_in1 Alias Portb.2
Motor_kanan_in2 Alias Portb.3
Motor_kiri_in1 Alias Portb.0
Motor_kiri_in2 Alias Portb.1

Config Motor_kanan_in1 = Output
Config Motor_kanan_in2 = Output
Config Motor_kiri_in1 = Output
Config Motor_kiri_in2 = Output

Config Porta.0 = Input
Config Porta.1 = Input
Config Porta.2 = Input
Config Porta.3 = Input
Config Portc.0 = Input
Config Portc.1 = Input

Scan_track:
  Gosub Baca_sensor_track
  Locate 1 , 1
  If Runstop = True Then
    Lcd Bin(data_track) ; " "
  Locate 2 , 1

  If Data_track = &B001100 Then                                'Lurus'
    Gosub Lurus
    Locate 2 , 14
    Lcd " ^ "
    Previus = Maju
    Status_track = Normal
  ElseIf Data_track = &B011110 Then                                'Lurus'
    Gosub Lurus
    Locate 2 , 14
    Lcd " ^ "
    Previus = Maju
    Status_track = Normal

```

Gambar 3.14. Script program Sensor *Proximity IR* dan Motor


```

Elseif Data_track = &B0010000 Then                                'Kiri'
  Gosub Belok_kiri_slow
  Locate 2 , 14
  Lcd "<"
  Status_track = Normal
  Previous = Kiri
Elseif Data_track = &B0111100 Then                                'Kiri'
  Gosub Belok_kiri_slow
  Locate 2 , 14
  Lcd "<"
  Status_track = Normal
  Previous = Kanan
Elseif Data_track = &B0110000 Then                                'Kiri'
  Gosub Belok_kiri_slow
  Locate 2 , 14
  Lcd "<"
  Status_track = Normal
  Previous = Kiri
Elseif Data_track = &B0100000 Then                                'Kiri'
  Gosub Belok_kiri_pol
  Locate 2 , 14
  Lcd "<"
  Status_track = Normal
  Previous = Kiri
Elseif Data_track = &B1100000 Then                                'Kiri'
  Gosub Belok_kiri_pol
  Locate 2 , 14
  Lcd "<"
  Status_track = Normal
  Previous = Kiri
Elseif Data_track = &B1000000 Then                                'Kiri'
  Gosub Belok_kiri_pol
  Locate 2 , 14
  Lcd "<"
  Status_track = Normal
  Previous = Kiri
Elseif Data_track = &B000100 Then                                  'Kanan'
  Gosub Belok_kanan_slow
  Locate 2 , 14
  Lcd ">"
  Status_track = Normal
  Previous = Kanan
Elseif Data_track = &B000110 Then                                  'Kanan'
  Gosub Belok_kanan_slow
  Locate 2 , 14
  Lcd ">"
  Status_track = Normal
  Previous = Kanan
Elseif Data_track = &B001110 Then                                  'Kanan'
  Gosub Belok_kanan_slow
  Locate 2 , 14
  Lcd ">"
  Status_track = Normal
  Previous = Kanan
Elseif Data_track = &B000010 Then                                  'Kanan'
  Gosub Belok_kanan_pol
  Locate 2 , 14
  Lcd ">"
  Status_track = Normal
  Previous = Kanan

```

Gambar 3.15. Script program Sensor *Proximity IR* dan Motor (lanjutan)

```

Elseif Data_track = &B0000111 Then                                'Kanan'
    Gosub Belok_kanan_pol
    Locate 2 , 14
    Lcd "> "
    Status_track = Normal
    Previous = Kanan
Elseif Data_track = &B0000001 Then                                'Kanan'
    Gosub Belok_kanan_pol
    Locate 2 , 14
    Lcd "> "
    Status_track = Normal
    Previous = Kanan

'_____Pertigaan Kanan_____ '
Elseif Data_track = &B0001111 Then
    Gosub Motor_stop
    Locate 2 , 14
    Lcd ">"
    Status_track = Pertigaan
Elseif Data_track = &B0011111 Then
    Gosub Motor_stop
    Locate 2 , 14
    Lcd ">"
    Status_track = Pertigaan
Elseif Data_track = &B0111111 Then
    Gosub Motor_stop
    Locate 2 , 14
    Lcd ">"
    Status_track = Pertigaan
Elseif Data_track = &B1011111 Then
    Gosub Motor_stop
    Locate 2 , 14
    Lcd ">"
    Status_track = Pertigaan

'_____Pertigaan Kiri_____ '
Elseif Data_track = &B1111100 Then
    Gosub Motor_stop
    Locate 2 , 14
    Lcd "<"
    Status_track = Pertigaan
Elseif Data_track = &B1111110 Then
    Gosub Motor_stop
    Locate 2 , 14
    Lcd "<"
    Status_track = Pertigaan
Elseif Data_track = &B1111000 Then
    Gosub Motor_stop
    Locate 2 , 14
    Lcd "<"
    Status_track = Pertigaan
Elseif Data_track = &B1111101 Then
    Gosub Motor_stop
    Locate 2 , 14
    Lcd "<"
    Status_track = Pertigaan

```

Gambar 3.16. Script program Sensor *Proximity IR* dan Motor (lanjutan)

```

' _____ Percabangan _____ '
Elseif Data_track = &B1111111 Then
  Gosub Motor_stop
  Locate 2 , 14
  Lcd "T "
  Status_track = Pertigaan
Elseif Data_track = &B001010 Then
  Gosub Lurus_slow
  Locate 2 , 14
  Lcd "<T "
  Status_track = Pertigaan
Elseif Data_track = &B010100 Then
  Gosub Lurus_slow
  Locate 2 , 14
  Lcd "<T "
  Status_track = Pertigaan
Elseif Data_track = &B110011 Then
  Gosub Lurus_slow
  Locate 2 , 14
  Lcd " T "
  Status_track = Pertigaan
Elseif Data_track = &B100111 Then
  Gosub Lurus_slow
  Locate 2 , 14
  Lcd " T "
  Status_track = Pertigaan
Elseif Data_track = &B111001 Then
  Gosub Lurus_slow
  Locate 2 , 14
  Lcd " T "
  Status_track = Pertigaan
Elseif Data_track = &B100011 Then
  Gosub Lurus_slow
  Locate 2 , 14
  Lcd " T "
  Status_track = Pertigaan
Elseif Data_track = &B110001 Then
  Gosub Lurus_slow
  Locate 2 , 14
  Lcd " T "
  Status_track = Pertigaan
Elseif Data_track = &B010001 Then
  Gosub Lurus_slow
  Locate 2 , 14
  Lcd " T "
  Status_track = Pertigaan
Elseif Data_track = &B100010 Then
  Gosub Lurus_slow
  Locate 2 , 14
  Lcd " T "
  Status_track = Pertigaan
Elseif Data_track = &B011011 Then
  Gosub Lurus_slow
  Locate 2 , 14
  Lcd " T "
  Status_track = Pertigaan
Elseif Data_track = &B110110 Then
  Gosub Lurus_slow
  Locate 2 , 14
  Lcd " T "
  Status_track = Pertigaan

```

Gambar 3.17. Script program Sensor *Proximity IR* dan Motor (lanjutan)

```

Elseif Data_track = &B011101 Then
    Gosub Lurus_slow
    Locate 2 , 14
    Lcd " T "
    Status_track = Pertigaan
Elseif Data_track = &B101110 Then
    Gosub Lurus_slow
    Locate 2 , 14
    Lcd " T "
    Status_track = Pertigaan
Elseif Data_track = &B010111 Then
    Gosub Lurus_slow
    Locate 2 , 14
    Lcd " T "
    Status_track = Pertigaan
Elseif Data_track = &B111010 Then
    Gosub Lurus_slow
    Locate 2 , 14
    Lcd " T "
    Status_track = Pertigaan
Elseif Data_track = &B010110 Then
    Gosub Lurus_slow
    Locate 2 , 14
    Lcd " T "
    Status_track = Pertigaan
Elseif Data_track = &B011010 Then
    Gosub Lurus_slow
    Locate 2 , 14
    Lcd " T "
    Status_track = Pertigaan
Elseif Data_track = &B111011 Then
    Gosub Lurus_slow
    Locate 2 , 14
    Lcd " T "
    Status_track = Pertigaan
Elseif Data_track = &B110111 Then
    Gosub Lurus_slow
    Locate 2 , 14
    Lcd " T "
    Status_track = Pertigaan
Elseif Data_track = &B010101 Then
    Gosub Lurus_slow
    Locate 2 , 14
    Lcd " T "
    Status_track = Pertigaan
Elseif Data_track = &B101010 Then
    Gosub Lurus_slow
    Locate 2 , 14
    Lcd " T "
    Status_track = Pertigaan
Elseif Data_track = &B010010 Then
    Gosub Lurus_slow
    Locate 2 , 14
    Lcd " T "
    Status_track = Pertigaan
Elseif Data_track = &B101101 Then
    Gosub Lurus_slow
    Locate 2 , 14
    Lcd " T "
    Status_track = Pertigaan

```

Gambar 3.18. Script program Sensor *Proximity IR* dan Motor (lanjutan)

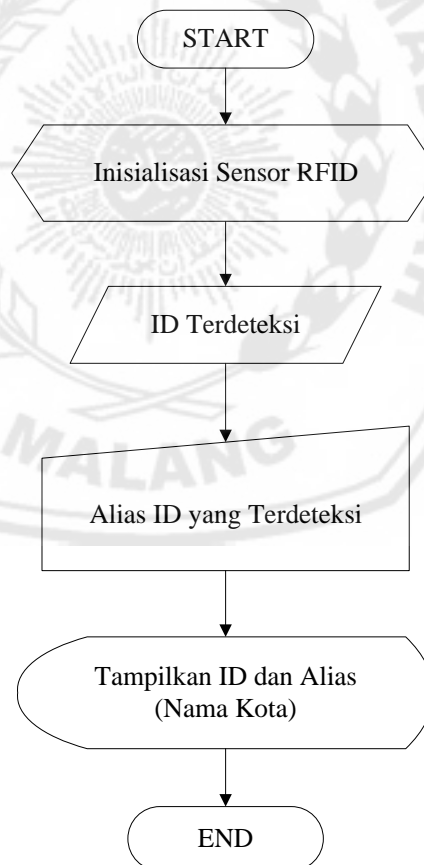
```

Elseif Data_track = &B0000000 Then
  Gosub Motor_stop
  Waitms 10
  Status_track = Normal
  If Previus = Kiri Then
    Gosub Belok_kiri_pol
    Gosub Baca_sensor_track
  Elseif Previus = Kanan Then
    Gosub Belok_kanan_pol
    Gosub Baca_sensor_track
  Elseif Previus = Maju Then
    Gosub Motor_mundur
    Gosub Baca_sensor_track
  End If
  Locate 2 , 14
  Lcd "out "
End If

```

Gambar 3.19. Script program Sensor *Proximity IR* dan Motor (lanjutan)

3.3.6. Radio Frequency Identification (RFID)



Gambar 3.20. Diagram Alir Sensor RFID (*Radio Frequency Identification*)

Pada gambar 3.20. flowchart RFID diawali inisialisasi sensor program RFID yang terhubung pada mikrokontroller. Setelah inisialisasi sensor RFID, selanjutnya sensor sudah dapat *scanning*, dengan cara memancarkan frekuensi radio. Setelah ID terdeteksi oleh sensor yang selanjutnya dibaca oleh mikrokontroller dan melakukan pencocokan Inisialisasi (Alias / Nama Kota) dengan ID yang telah ditentukan untuk tiap kota. Untuk mengetahui ID yang dipakai dan nama-nama kota dapat dilihat pada tabel 3.4. berikut,

Tabel 3.4. Alias (nama kota) pada ID yang digunakan

Urutan	Nomor ID	Nama Kota
1	1C00974F14D0	Aceh
2	1C00967250A8	Blangkarajen
3	1D00361F0632	Medan
4	1D00361F1125	Pekanbaru
5	1D003869B5F9	Bukit Tinggi
6	1C009356EF36	Padang
7	27004915F982	Bengkulu
8	2700498612FA	Jambi
9	27004915F08B	Palembang
10	2700490DB0D3	Lampung

Dapat dilihat pada tabel 3.4. setiap kota memiliki ID yang berbeda beda, untuk pengiriman data lokasi juga memerlukan Inisialisasi. Data yang dikirimkan oleh robot hanya berupa nomor urut yang telah ditentukan sebelumnya. Hal ini di peruntukan supaya data yang dikirimkan bisa lebih ringkas. Untuk *list program* dari sensor RFID dapat dilihat pada gambar 3.21 dan gambar 3.22. berikut,

```

$regfile = "m1284pdef.dat"
$crystal = 8000000
$baud = 9600
$baudl = 9600

    Open "COM1:" For Binary As #1
    Open "COM2:" For Binary As #2

Aceh Alias "1C00974F14D0"
Blangkarajen Alias "1C00967250A8"
Medan Alias "1D00361F0632"
Pekanbaru Alias "1D00361F1125"
Bukittinggi Alias "1D003869B5F9"
Padang Alias "1C009356EF36"
Bengkulu Alias "27004915F982"
Jambi Alias "2700498612FA"
Palembang Alias "27004915F08B"
Lampung Alias "2700490DB0D3"

    On Urxc1 Proses_hcl2
    On Urxc Rec_isr
Enable Urxc

Dim Datafix As Integer
Dim Dataok As Single
Dim Strdatok As String * 5
Dim Dae As Integer , A As Byte
Dim Data1 As Integer
Dim Data2 As Integer , Temp As Single
Dim Buf As String * 50 , D As Byte , Ser_flag As Bit , Los As Bit
Dim Id As String * 15 , Mode_id As Integer , Last_mode As Integer
Dim Posisi As Integer , Last_posisi As Integer , Kota_asal As Integer
Dim Kota_tujuan1 As Integer , Kota_tujuan2 As Integer , Kota_finish As Integer
Dim Status_track As Integer , Last_id As String * 15
Dim S As String * 50 , B As Byte , Newdata As Bit , Statuscekkota1 As Bit , Statuscekkota2 As Bit

Scan_id_track:
    If Ser_flag = 1 Then
        Reset Ser_flag
        Los = 0
        If Last_id <> Id Then
            Last_posisi = Posisi
            Last_id = Id
        Gosub Motor_stop
        End If
        Locate 1 , 1
        Lcd Id ; " "
        Locate 2 , 1
        If Id = Aceh Then
            Lcd "ACEH"
            Posisi = Id_aceh
        ElseIf Id = Blangkarajen Then
            Lcd "BLANGKARAJEN"
            Posisi = Id_blangkarajen
        ElseIf Id = Medan Then
            Lcd "MEDAN"
            Posisi = Id_medan
        ElseIf Id = Pekanbaru Then
            Lcd "PEKANBARU"
            Posisi = Id_pekanbaru

```

Gambar 3.21. Script program Sensor RFID

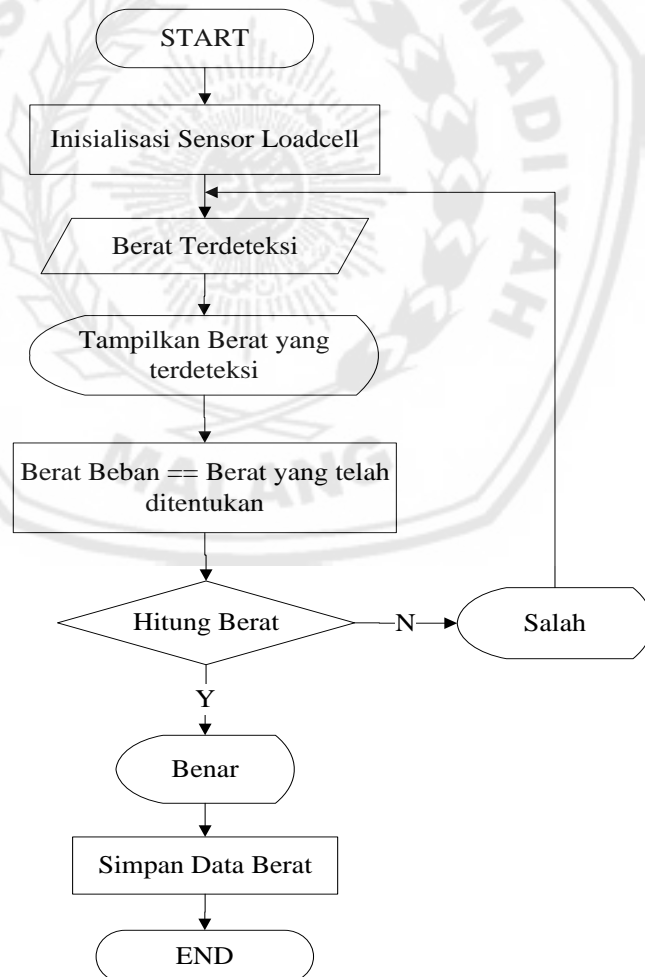

```

Elseif Id = Bukittinggi Then
  Lcd "BUKITTINGGI "
  Posisi = Id_bukittinggi
Elseif Id = Padang Then
  Lcd "PADANG "
  Posisi = Id_padang
Elseif Id = Bengkulu Then
  Lcd "BENGKULU "
  Posisi = Id_bengkulu
Elseif Id = Jambi Then
  Lcd "JAMBI "
  Posisi = Id_jambi
Elseif Id = Palembang Then
  Lcd "PALEMBANG "
  Posisi = Id_palembang
Elseif Id = Lampung Then
  Lcd "LAMPUNG "
  Posisi = Id_lampung
End If

```

Gambar 3.22. Script program Sensor RFID (lanjutan)

3.3.7. Loadcell



Gambar 3.23. Diagram Alir Sensor Loadcell

Pada gambar 3.23. flowchart *Loadcell* diawali inisialisasi program atau pengenalan pada mikrokontroller dengan sensor *Loadcell* apakah telah terhubung. Setelah inisialisasi sensor *Loadcell* menunggu inputan yang berupa objek yang mempunyai masa (bobot) yang telah ditentukan. Setelah masa objek terdeteksi oleh sensor *Loadcell*, maka sensor akan mengirimkan data ke Mikrokontroler untuk pengecekan kebenaran bobot dari objek yang dimasukan / diangkut. Untuk *list program* dari sensor *Loadcell* dapat dilihat pada gambar 3.24. dan gambar 3.25. berikut,

```

$regfile = "m1284pdef.dat"
$crystal = 8000000
$baud = 9600
$baudl = 9600

Adsk Alias Porta.4 ' use pin D.5 as output sck
Addo Alias Pina.5 ' use pin D.6 as input Dataout
  Config Adsk = Output
  Config Addo = Input
Dim Count As Long
Dim I As Byte
Dim Hx711_read As Single
Dim Hx711 As String * 10
Dim Datafix As Integer
Dim Dataok As Single
Dim Strdatok As String * 5
Dim Dae As Integer , A As Byte
Dim Data1 As Integer
Dim Data2 As Integer , Temp As Single
Dim Berat As Single , Buf_hc As String * 50 , Dhc As Byte
Dim Buf As String * 50 , D As Byte , Ser_flag As Bit , Los As
  Bit
Dim S As String * 50 , B As Byte , Newdata As Bit ,
  Statuscekkota1 As Bit , Statuscekkota2 As Bit
Dim Databerat As Integer , Count1 As Integer , Temps As String
  * 5 , Runstop As Bit

  If Kota_tujuan1 = Posisi And Statuscekkota1 = True Then
    Upperline
    Lcd "Sampai di kota 1"
    Gosub Motor_stop
    Waitms 200
    Cls
    Lcd "Masukkan Oleh2 1"
    Do
      Gosub Baca_loadcell
      If Berat > 1 Then
        If Berat >= 25 And Berat <= 50 Then
          Lowerline
          Lcd "BENAR"
          Waitms 55
          Los = 0
          Last_mode = 2
          Statuscekkota1 = False
          Exit Do
        
```

Gambar 3.24. Script program Sensor *Loadcell*

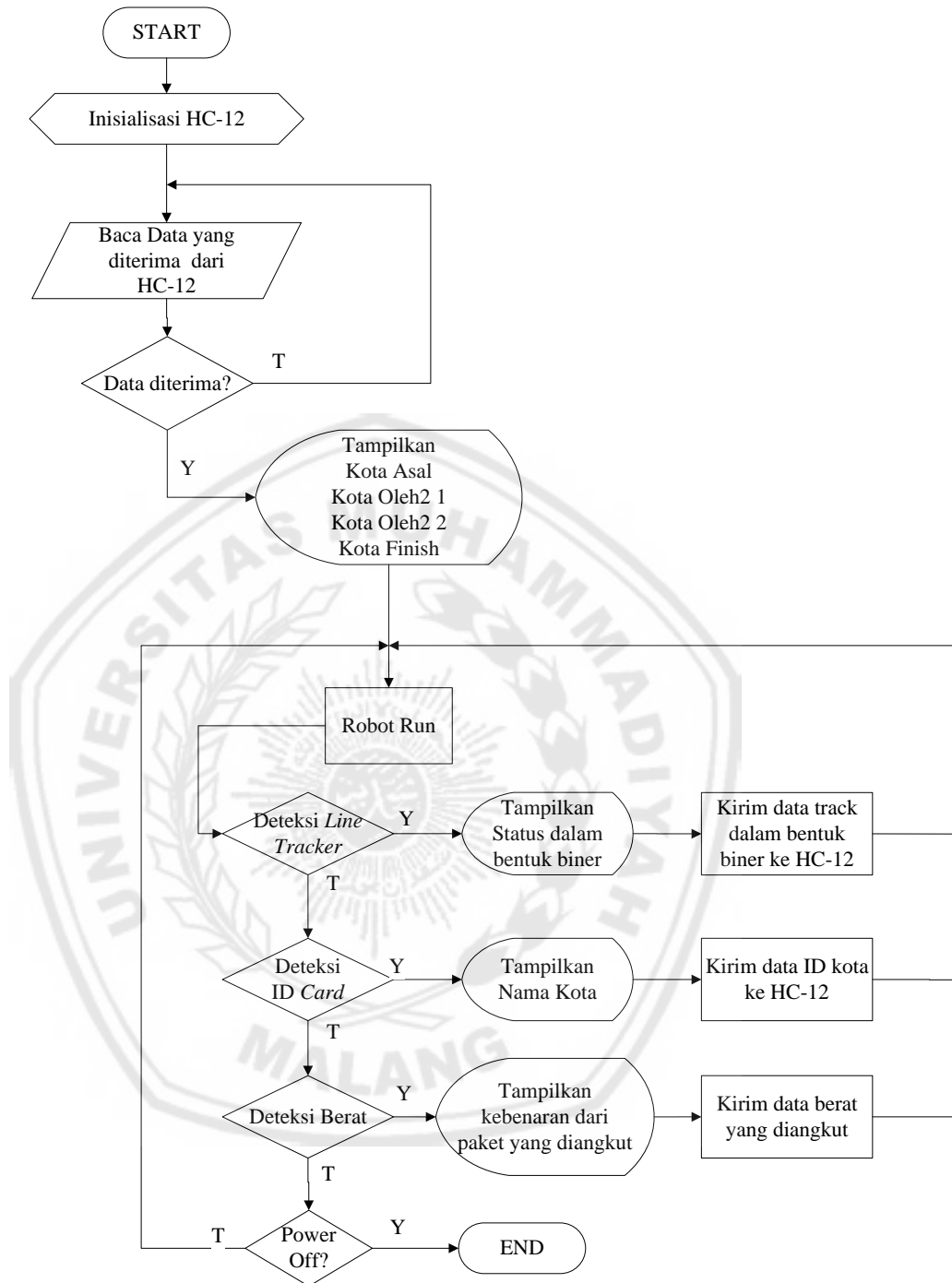
```

        Else
            Lowerline
            Lcd "SALAH"
        End If
    Else
        Lowerline
        Lcd Space(16)
    End If
Loop
End If
If Kota_tujuan2 = Posisi And Statuscekkota2 = True Then
    Upperline
    Lcd "Sampai di kota 2"
    Cls
    Lcd "Masukkan 0leh2 2"
    Gosub Motor_stop
    Wait 1
    Do
        Gosub Baca_loadcell
        If Berat > 1 Then
            If Berat > 50 Then
                Lowerline
                Lcd "BENAR"
                Waitms 55
                Los = 0
                Last_mode = 2
                Statuscekkota2 = False
                Exit Do
            Else
                Lowerline
                Lcd "SALAH"
            End If
        Else
            Lowerline
            Lcd Space(16)
        End If
    Loop
End If
If Kota_finish = Posisi Then
    Cls
    Runstop = False
    Statuscekkota1 = False
    Statuscekkota2 = False
End If
Wait 1
End If
Return

```

Gambar 3.25. Diagram Alir Sensor *Loadcell* (lanjutan)

3.3.8. Wireless Pada Robot



Gambar 3.26. Diagram Alir *Wireless* HC-12 pada Robot

Dilihat dari gambar 3.26. merupakan diagram alir dari *wireless* pada robot. Yang mana *wireless* pada robot berperan sebagai penerima data dari aplikasi dan pengiriman data ke aplikasi. Saat penerimaan data dari aplikasi robot hanya membaca dan menampilkan data yang telah diterima oleh

wireless yaitu data Lokasi (Kota Awal, Kota Tujuan dan Kota pengangkutan). Sedangkan saat pengiriman data ke aplikasi robot, data yang dikirimkan adalah data lokasi, data berat, dan data track. Untuk *listing program wireless* robot dapat dilihat pada gambar 3.27. dan gambar 3.28. berikut,

```

$regfile = "m1284pdef.dat"
$crystal = 8000000
$baud = 9600
$baudl = 9600

$hwstack = 255
$swstack = 255
$framesize = 355

Declare Sub Getline_hcl2(s As String)
Open "COM1:" For Binary As #1
Open "COM2:" For Binary As #2
Set_hc Alias Portd.4
Config Set_hc = Output

    Dim Datafix As Integer
    Dim Dataok As Single
    Dim Strdatok As String * 5
    Dim Dae As Integer , A As Byte
    Dim Data1 As Integer
    Dim Data2 As Integer , Temp As Single

Setup_hcl2:
    Reset Set_hc
    Cls
    Lcd "SETUP Wi-Fi"
    Lowerline
    Waitms 250
    Print #2 , "AT"
    Getline_hcl2 Data_hc
    Lcd ":" ; Data_hc ; "      "
    Locate 2 , 1

    Print #2 , "AT+UBN1"
    Getline_hcl2 Data_hc
    Lcd ":" ; Data_hc ; "      "
    Locate 2 , 1

    Print #2 , "AT+COL1"
    Getline_hcl2 Data_hc
    Lcd ":" ; Data_hc ; "      "
    Set Set_hc
    Wait 1
Return

```

Gambar 3.27. Script program *Wireless* HC-12 pada Robot

```

Proses_hcl2:
  Disable Urxc1
  Dhc = Udr1
  If Chr(dhc) = ":" Then Buf_hc = ""
  If Dhc <> 0 And Dhc <> 13 And Dhc <> 10 Then Buf_hc = Buf_hc +
    Chr(dhc) 'add to buffer
  If Dhc = 10 Then
    Data_hc = Buf_hc
    Newdata = True
  End If
  Enable Urxc1
Return

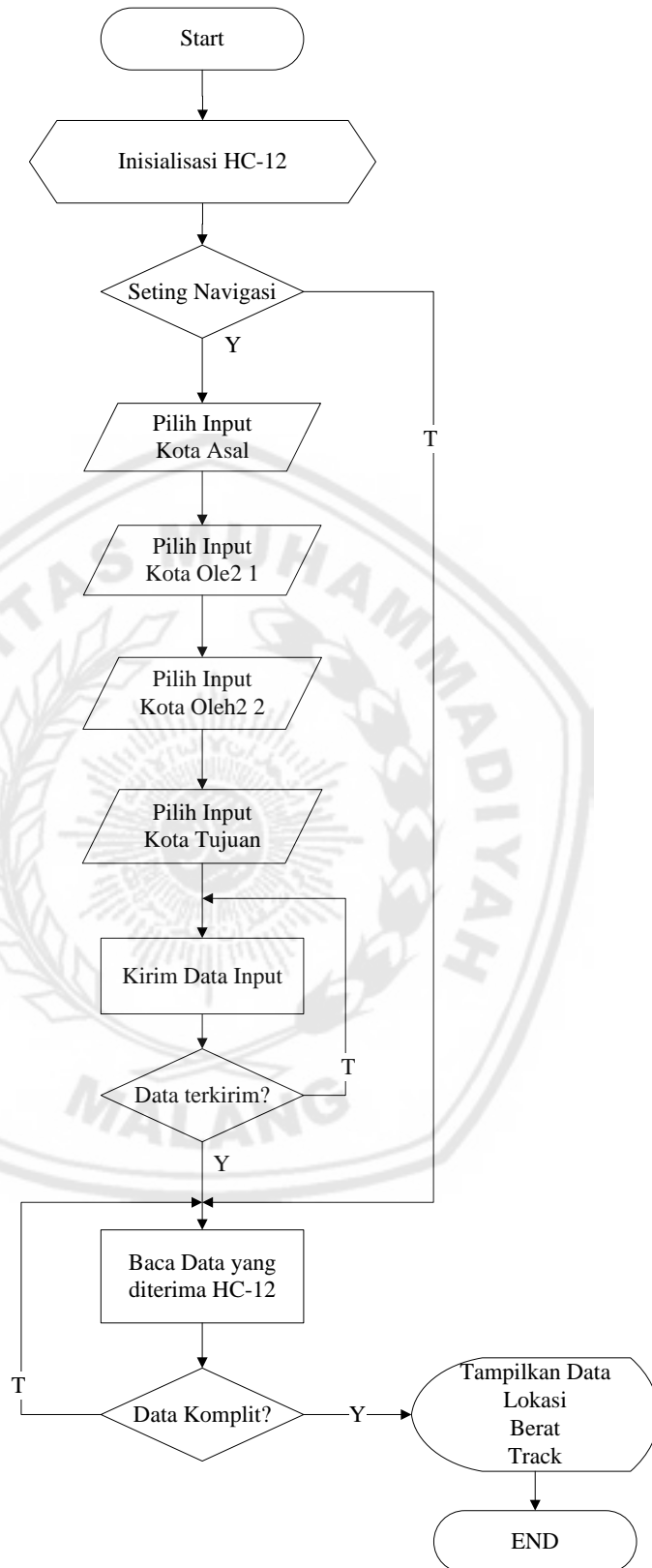
Rec_isr:
  Disable Urxc
  D = Udr
  If D = 2 Then Buf = ""
  If Los = 0 Then
    If D <> 3 And D <> 2 Then Buf = Buf + Chr(d)
    D = 3 Then
      Los = 1
      Waitms 100
      Set Ser_flag
      Id = Buf
      Gosub Motor_stop
    End If
  End If
  Enable Urxc
Return

Sub Getline_hcl2(s As String)
  S = ""
  A = Ischarwaiting(#2)
  If A = 1 Then
    Do
      B = Waitkey(#2)
      Select Case B
        Case 0
        Case 13
        Case 10 : If S <> "" Then Exit Do
        Case Else
          S = S + Chr(b)
        End Select
    Loop
  End If
End Sub

```

Gambar 3.28. Diagram Alir Wireless HC-12 pada Robot (lanjutan)

3.3.9. Wireless Aplikasi



Gambar 3.29. Diagram Alir pada *wireless* aplikasi

Pada gambar 3.29. menunjukan bahwa diagram alir dari *wireless* yang ada pada aplikasi hanya berperan sebagai pengirim dan menerima data. Untuk data yang dikirimkan berupa data lokasi kota. Sedangkan data yang di terima berupa data lokasi yang sedang disinggahi, data berat barang / paket, dan data *track*. Untuk data yang dikirimkan ada empat data yang perlu di *input*, ke-empat data tersebut meliputi kota awal, kota pengangkutan (oleh-oleh 1 & Oleh-oleh 2), dan kota tujuan. Untuk *listing program wireless* dapat dilihat pada gambar 3.30. dan gambar 3.34. berikut,

```
unit Input;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, StdCtrls, jpeg, ExtCtrls, ComCtrls, ActnMan,
  ActnColorMaps,
  SUIButton, SUIComboBox, SUIImagePanel, registry, VaClasses,
  VaComm,
  SUIStatusBar, SUIForm;

type
  TForm1 = class(TForm)
    Send: TButton;
    Shape1: TShape;
    suiPanel2: TsuiPanel;
    Label7: TLabel;
    Label4: TLabel;

    suiPanel1: TsuiPanel;
    sumatra: TImage;
    suiPanel3: TsuiPanel;
    Labelp: TLabel;
    comboport: TsuiComboBox;

    suiButton7: TsuiButton;
    Button1: TsuiButton;
    suiPanel4: TsuiPanel;
    Kota_Awal: TComboBox;
    Buah_Tangan: TComboBox;
    Buah_tangan02: TComboBox;

    Finish: TComboBox;
    suiPanel5: TsuiPanel;
    berat: TLabel;
    Timer2: TTimer;
    VaComm1: TVaComm;

    suiStatusBar1: TsuiStatusBar;
    suiForm1: TsuiForm;
    Button2: TButton;
    Edit1: TEdit;
    Memo1: TMemo;
  end;

```

Gambar 3.29. Script program *wireless* aplikasi


```

Button3: TButton;
Button4: TButton;
suiPanel6: TsuiPanel;
Label11: TLabel;
Label15: TLabel;

suiPanel7: TsuiPanel;
lokasi: TLabel;
suiPanel8: TsuiPanel;
status: TLabel;
Button5: TButton;

Label2: TLabel;
Label3: TLabel;
Label6: TLabel;
Label8: TLabel;

procedure suiButton7Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Timer2Timer(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure VaComm1RxChar(Sender: TObject; Count: Integer);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure SendClick(Sender: TObject);
procedure Button5Click(Sender: TObject);

private
{ Private declarations }
public
{ Public declarations }
    procedure cek_port;
    function scan_id_lokasi(scan:string):string;
end;

const
Id_aceh = '1';
Id_blangkarajen = '2';
Id_medan = '3';
Id_pekanbaru = '4';
Id_bukittinggi = '5';
Id_padang = '6';
Id_bengkulu = '7';
Id_jambi = '8';
Id_palembang = '9';
Id_lampung = '10';

implementation

{$R *.dfm}

procedure TForm1.cek_port;
var Reg : TRegistry;
    i:integer;
begin
    comboport.Clear;
    Reg := TRegistry.Create;
    reg.RootKey:=HKEY_LOCAL_MACHINE;
    try
        if Reg.OpenKeyReadOnly('\HARDWARE\DEVICEMAP\SERIALCOMM')=
True then

```

Gambar 3.30. Script program wireless aplikasi (lanjutan)


```

begin
for i:=0 to 30 do
begin
if Reg.ValueExists('\Device\Serial'+inttostr(i)) then
begin
Comboport.Items.Add(
Reg.ReadString('\Device\Serial'+inttostr(i)));
comboport.ItemIndex:=0;
comboport.Refresh;
end;
end;
for i:=0 to 30 do
begin
if Reg.ValueExists('\Device\Silabser'+inttostr(i)) then
begin
Comboport.Items.Add(
Reg.ReadString('\Device\Silabser'+inttostr(i)));
comboport.ItemIndex:=0;
comboport.Refresh;
end;
end;
if Reg.ValueExists('\Device\USBSE000') then
begin
Comboport.Items.Add(
Reg.ReadString('\Device\USBSE000'));
comboport.ItemIndex:=0;
comboport.Refresh;
end;
Reg.CloseKey;
end;
finally
Reg.Free;
inherited;
end;
end;

procedure TForm1.suiButton7Click(Sender: TObject);
begin
cek_port;
end;

procedure TForm1.Button1Click(Sender: TObject);
var
waktu: tDateTime;
temp,port: string;
com,code:integer;
begin
if button1.Caption='Connect' then
begin
port:=comboport.Items.Strings[comboport.itemindex];
if port<>'' then
begin
vacomml.Close;
temp:=port;
// vacomml.DeviceName:=port;
if com>9 then
begin
vacomml.DeviceName:='\.\'+port;
end else

```

Gambar 3.31. Script program wireless aplikasi (lanjutan)

```

begin
    delete(temp1,3);
    val(temp1,com,i);
    vacomml.PortNum:=com;
    vacomml.DeviceName:=port;
end;
vacomml.Open;
button1.Caption:='Disconnect';
konek:=true;
suiStatusBar1.Panels[0].Text      := 'Koneksi      pada
'+port+'.....';
vacomml.SetDTR(true);    //mode set HC
Sleep(800);
mode_HC:=true;
vacomml.WriteText('AT'+#13+#10);
end;
end else
begin
    konek:=false;
    sleep(200);
    vacomml.Close;
    button1.Caption:='Connect';
    suiStatusBar1.Panels[0].Text := 'Disconnect';
    statuscom:=false;
    showmessage('Koneksi diputus');
end;
end;
procedure TForm1.FormCreate(Sender: TObject);
begin
    cek_port;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
    vacomml.WriteText(edit1.Text+#13+#10);
end;
function tform1.scan_id_lokasi(scan:string):string;
begin
    if scan=id_aceh then
    begin
        result:='ACEH';
    end else if scan=id_blangkarajen then
    begin
        result:='BLANGKARAJEN';
    end else if scan=id_medan then
    begin
        result:='MEDAN';
    end else if scan=id_pekanbaru then
    begin
        result:='PEKANBARU';
    end else if scan=id_bukittinggi then
    begin
        result:='BUKIT TINGGI';
    end else if scan=id_padang then
    begin
        result:='PADANG';
    end else if scan=id_bengkulu then
    begin
        result:='BENGKULU';
    end else if scan=id_jambi then
    begin
        result:='JAMBI';
    end;
end;

```

Gambar 3.32. Script program wireless aplikasi (lanjutan)

```

        end else if scan=id_palembang then
        begin
            result:='PALEMBANG';
        end else if scan=id_lampung then
        begin
            result:='LAMPUNG';
        end else
        begin
            result:='Tidak Terdeteksi';
        end;
    end;

procedure TForm1.VaComm1RxChar(Sender: TObject; Count: Integer);
var scan:string;

begin
    data:=data+vacomm1.ReadText;
    if pos('#13#10,data)>1 then
    begin
        result:=copy(data,1,pos('#13,data)-1);
        if (mode_HC=true) and (result='OK') then
        begin
            vacomm1.WriteText('AT+U&N1'+#13+#10);
        end else if (mode_HC=true) and (result='OK+U&N1') then
        begin
            vacomm1.WriteText('AT+C011'+#13+#10);
        end else if (mode_HC=true) and (result='OK+C011') then
        begin
            mode_HC:=false;
            vacomm1.SetDTR(false); //mode data HC
            suiStatusBar1.Panels[0].Text := 'Koneksi OK pada
'+comboport.Items.Text+'.....';
        end else
        begin
            //data slave
            if (mode_HC=false) and (length(result)>=15) then
            begin
                memo1.Lines.Add(result);
                delete(result,1,pos(':',result));
                scan:=copy(result,1,pos('|',result)-1); //data awal
                lokasi.Caption:=scan_id_lokasi(scan);
                delete(result,1,pos('|',result));
                scan:=copy(result,1,pos('|',result)-1); //data berat
                berat.Caption:=scan;
                delete(result,1,pos('|',result));
                scan:=copy(result,1,pos('|',result)-1); //data sensor
                delete(scan,1,2);
                status.Caption:=scan;
            end;
        end;
        data:='';
    end;
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
    vacomm1.SetDTR(true);
end;

```

Gambar 3.33. Script program wireless aplikasi (lanjutan)

```

procedure TForm1.Button4Click(Sender: TObject);
begin
    vacomml.SetDTR(false);

    end;

procedure TForm1.SendClick(Sender: TObject);
begin
    if (kota_awal.ItemIndex=-1) or (buah_tangan.ItemIndex=-1) or (buah_tangan02.ItemIndex=-1) or (finish.ItemIndex=-1) then
    begin
        showmessage('Daftar Kota belum diset!!!');
    end else
    begin
        kotaawal:=kota_awal.ItemIndex+1;
        oleholeh1:=buah_tangan.ItemIndex+1;
        oleholeh2:=buah_tangan02.ItemIndex+1;
        tujuan:=finish.ItemIndex+1;
        protokol:='' + inttostr(kotaawal) + '|';
        protokol:=protokol + inttostr(oleholeh1) + '|';
        protokol:=protokol + inttostr(oleholeh2) + '|';
        protokol:=protokol + inttostr(tujuan) + '|';
        // showmessage(protokol);
        vacomml.WriteText(protokol+#13+#10);
    end;
end;

procedure TForm1.Button5Click(Sender: TObject);
begin
    memol.Clear;
end;

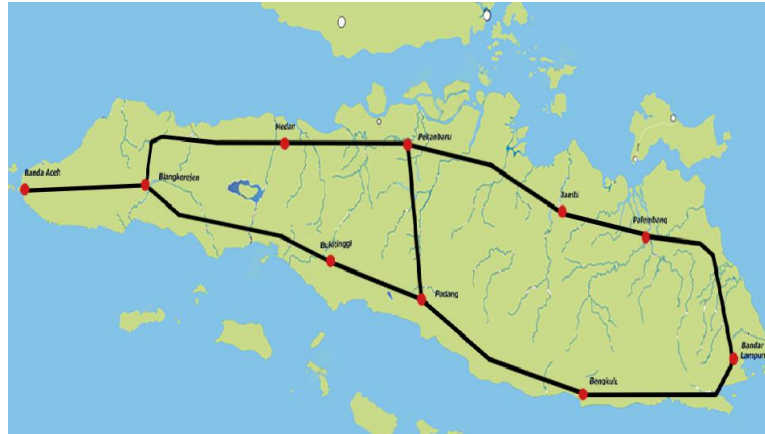
end.

```

Gambar 3.34. Script program wireless aplikasi (lanjutan)

3.3.10. Perencanaan Jalur

Peta pulau Sumatra ini menggunakan 10 titik kota pemberhentian (kota besar). Pada peta pulau Sumatra terdapat jalur transportasi yang digunakan sebagai jalur Robot *Line Follower*. Di jalur tersebut tempat dimana robot akan mengikuti jalur garis yang berwarna gelap (hitam). Terdapat tiga percabangan dalam jalur robot, untuk dapat membedakan jalur tersebut robot akan memanfaatkan ID kota. Gambar peta pulau Sumatra ini di buat pada *form designer*. Gambar peta pulau Sumatra diperlihatkan pada gambar 3.35.

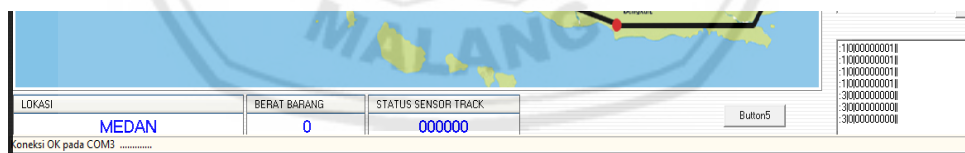


Gambar 3.35 *Form 1 Jalur pada Peta Pulau Sumatra*

Dalam gambar 3.35. terdapat 10 titik yang berwarna merah. Kesepuluh titik ini merupakan lokasi Kota, sekaligus penempatan dimana koordinat ID Kota itu berada. Serta terdapat tiga percabangan yang ada di kota Blangkarajen, Pekanbaru, dan Padang. Untuk percabangan juga masih menggunakan ID Kota yang ada pada posisinya.

3.3.11. Perencanaan Petunjuk Lokasi Robot

Pada perancangan petunjuk lokasi pergerakan robot ini menyesuaikan koordinat dari titik-titik kota yang ada pada peta. Untuk sample kolom penunjuk posisi robot dapat dilihat pada gambar 3.36.



Gambar 3.36. Kolom penunjuk posisi robot dan kondisi robot

Pada gambar 3.36. terdapat tiga kolom, yang mana ke-tiga kolom tersebut berperan sebagai monitoring dari robot. Kolom pertama berperan sebagai penunjuk lokasi robot berada. Kolom ke-dua berperan sebagai *monitoring* dari berat barang yang diangkut. Dan pada kolom ke-tiga yang merupakan “status sensor track” berperan sebagai monitoring dari sensor *Proximity Ir*.